

Programa del curso IC7841

Proyecto de Ingeniería de Software

Escuela de Computación

Carrera de Ingeniería de Computación, Plan 411.

I Parte. Aspectos relativos al plan de estudios

1 Datos generales

Nombre del curso:	Proyecto de Ingeniería de Software
Código:	IC7841
Tipo de curso:	Seminario
Nº de créditos:	3
Nº horas de clase por semana:	4
Nº horas extraclase por semana:	5
Requisitos:	IC4302 Bases de Datos 2 IC6831 Aseguramiento de la Calidad
Correquisitos:	Ninguno
El curso es requisito de:	IC8842 Práctica profesional
Asistencia:	Obligatoria
Suficiencia:	No
Posibilidad de reconocimiento:	Sí
Vigencia del programa:	II semestre 2024.

2 Descripción General

El curso introduce los conceptos y técnicas necesarios para la creación de productos con tecnologías de tendencia reciente, así como los impactos arquitectónicos que estos puedan tener. El curso presta atención al enfoque de diseño centrado en el usuario, atendiendo especialmente lo concerniente al diseño de la experiencia del usuario. Los estudiantes trabajan en equipos que siguen un enfoque ágil que integra procesos, herramientas y técnicas de análisis, diseño, construcción, verificación y validación de software aprendidos a lo largo de la carrera, para lograr un producto expresado en tecnologías modernas que componen una arquitectura adecuada.

Aplica de manera práctica un proceso de desarrollo de software completo, pasando por todas las etapas – desde el estudio del problema y las necesidades de un cliente real, pasando por el diseño de una solución informática, hasta construir, integrar y probar el producto. Se desarrolla un sistema planteado por los estudiantes del curso, en grupos de 2 o 3 estudiantes.

Este curso resume e integra las mejores practicas en el desarrollo de sistemas y prepara al estudiante en una de las actividades que puede llegar e desarrollar en el futuro inmediato como profesional en Ingeniería de Software

3 Objetivos

Objetivo General

Desarrollar un sistema que integre conocimientos y mejores prácticas como profesional en Ingeniería de Software.

Objetivos Específicos

- Gestionar un proyecto de desarrollo de software siguiendo un enfoque ágil en el proceso que enmarque las actividades técnicas.
- Analizar un problema de desarrollo de software e modelar el dominio en el cual se circunscribe.
- Describir textualmente los problemas y los requerimientos, tomando en cuenta el contexto, dominio, y limitaciones/restricciones que presenten los usuarios y los interesados.
- Modelar las clases, las entidades, las relaciones, las operaciones y los comportamientos claves del dominio del problema.
- Diseñar una arquitectura para solucionar el problema, evaluando alternativas que busquen favorecer los principales atributos de calidad identificados como requerimientos no-funcionales.
- Aplicar patrones apropiados en los distintos niveles de diseño de la solución.
- Especificar los componentes de la arquitectura de software propuesta.
- Construir programas y componentes para satisfacer sus especificacio-

nes.

- Realizar revisiones sobre documentos técnicos y de código de programación.
- Planear, diseñar y aplicar pruebas a las unidades, componentes y subsistemas, en todos niveles requeridos, para evaluar su funcionamiento.
- Integrar los elementos y probar su integración para ensamblar un sistema completo de software que satisfaga los requerimientos según los alcances establecidos.
- Trabajar en equipos de manera colaborativa, para lograr un producto de alta calidad al final del proyecto.
- Producir documentación técnica de la especificación, el análisis, el diseño, los componentes, las pruebas y los manuales de usuario.
- Mantener los documentos técnicos y administrativos que permitan dar visibilidad al trabajo del equipo, controlar sus avances y atender los riesgos.
- Gestionar rigurosamente las versiones y administrar la configuración de los elementos de software, la documentación y demás recursos que se generen como parte del proyecto.

4 Contenidos

- Enfoques ágiles para el desarrollo de software
- Definición de los usuarios y dominios para el desarrollo del proyecto
- Repertorio de Tecnologías de software
 - Tecnologías de software para el desarrollo web
 - Tecnologías de software para aplicaciones con dispositivos móviles
 - Tecnologías de software para construcción de servicios distribuidos y cómputo en la nube
- Diseño de experiencia del usuario
 - Principios de la experiencia del usuario
 - Arquitectura de la información
 - Paradigmas de interacción
 - Diseño de experiencias de usuario
- Arquitecturas de aplicaciones de software
 - Estilos y patrones arquitectónicos
 - Arquitecturas multi-capas
 - Arquitecturas de aplicaciones web
 - Servicios de arquitecturas orientadas a servicios
 - Arquitecturas con dispositivos móviles
 - Otras consideraciones entre arquitectura y tecnología
- Pruebas de software
- Trascición e implantación del producto
 - Integración y prueba final del producto
 - Construcción de ayudas en línea
 - Manuales de usuario
 - Manuales técnicos
 - Capacitación de usuario
- Gestión de calidad del software
- Gestión de la configuración

II Parte. Aspectos operativos

5 Metodología de enseñanza y aprendizaje

La metodología del curso consistirá en la combinación de clases magistrales por parte del profesor, sesiones de seguimiento grupal, exposiciones de avance y de investigaciones, y el trabajo de campo de los estudiantes con sus usuarios. Las tecnologías y arquitecturas por usar en el curso podrán ser variadas periódicamente, según determine el Consejo de la Escuela de Ingeniería en Computación. Al inicio del curso se definirán los dominios sobre los cuales los estudiantes realizarán el proyecto. Durante el proyecto los estudiantes seleccionarán algún tema de interés o problema que hayan encontrado durante el desarrollo del proyecto y redactarán un breve artículo al respecto.

El curso trata de la aplicación práctica de un proceso ágil de desarrollo de software para desarrollar programas útiles para usuarios reales. Se brindan los conceptos tecnológicos y metodológicos que complementan el bagaje de conocimientos de los estudiantes, para que puedan construir un producto de software con tecnologías de tendencia reciente. El proceso de software abarca desde el análisis de un problema hasta la implementación de su solución. El curso requiere que los estudiantes tengan una gran capacidad de investigación y aprendizaje autónomo. El profesor les guía en ese proceso, pero es esperable que haya mucha diversidad tecnológica y el profesor no será experto en cada posible tecnología por utilizar. El proyecto por desarrollar deber tener un alcance definido que no sobrepase las 16 semanas de trabajo.

6 Evaluación

A continuación se detalla la evaluación del curso:

	%
Anteproyecto	10
Exposiciones	10
Planificación del proyecto	30
Ejecución del proyecto	30
Artículo	20
	100

El proyecto se dividirá en 3 entregables o **Sprints**, cada sprint termina con la entrega de tres productos:

- Un **Producto Mínimo Viable** (MVP) del sistema, el cual debe contemplar funcionalidades completas de una sección del sistema, desde la

persistencia en base de datos, pasando por la lógica de negocio, hasta la interfaz gráfica.

- Un **Manual de usuario** que incorpore todas las funcionalidades de los sprints previos y del actual con capturas de pantalla e instrucciones orientadas al usuario final.
- Un **Manual Técnico** basado en la plantilla provista en el material del curso. El Manual Técnico debe contemplar instrucciones para instalar tanto el entorno de desarrollo como el sistema en producción, en el entendido que el sistema siempre debe ser un Producto Viable.

La calificación del Sprint se divide en dos etapas. La etapa de Planificación (10 puntos por sprint) se enfoca en el análisis detallado y diseño del sistema. La etapa de Ejecución (10 puntos por sprint) se enfoca en el desarrollo del producto, gestión de la configuración, actividades de control, y documentación de apoyo. Al final de cada sprint se evaluará la calidad del MVP con una nota global de 0 a 100 que dependerá de la completitud del MVP y de la calidad del producto, esta calificación se multiplicará por la nota de las dos etapas correspondientes a ese sprint, siguiendo esta fórmula:

$$Planificacion = Planificacion * (NotaMVP/100)$$

$$Ejecucion = Ejecucion * (NotaMVP/100)$$

Esto implica que si el MVP no funciona, o si presenta funcionalidades incompletas, la nota de todo el sprint se verá afectada.

7 Plataforma del curso

Los materiales y asignaciones del curso se presentarán en la plataforma digital Canvas, se enviará una invitación directa a cada participante en la primera semana del curso. La comunicación oficial se hará por medio de la plataforma Discord, en ella deberá ingresar y asociarse los roles respectivos para acceder a la sección privada del curso. <https://discord.gg/ZKWtxDUQBs>.

8 Bibliografía

- Kniberg, H. (2007). Scrum y XP desde las trincheras. C4Media.
- Poppendieck, M. B., & Poppendieck, T. D. (2013). Lean software development: an agile toolkit. Boston: Addison Wesley.
- Norman, D. (2013). The design of Everyday Things. Nueva York: Basic Books.

9 Profesor

Bach. Rodolfo Mora Zamora.

Email: rodmora@itcr.ac.cr

Telegram: @Rjmoraza

Consulta: Martes y Jueves de 10am a 12md.